GOMS

Adapted from Berkeley Guir
& Caitlin Kelleher

GOMS

- Goals – what the user wants to do
- Operators – actions performed to reach the goal
- Methods – sequences of operators that accomplish a goal
- Selection Rules – describe when to choose one method over another

What is GOMS?

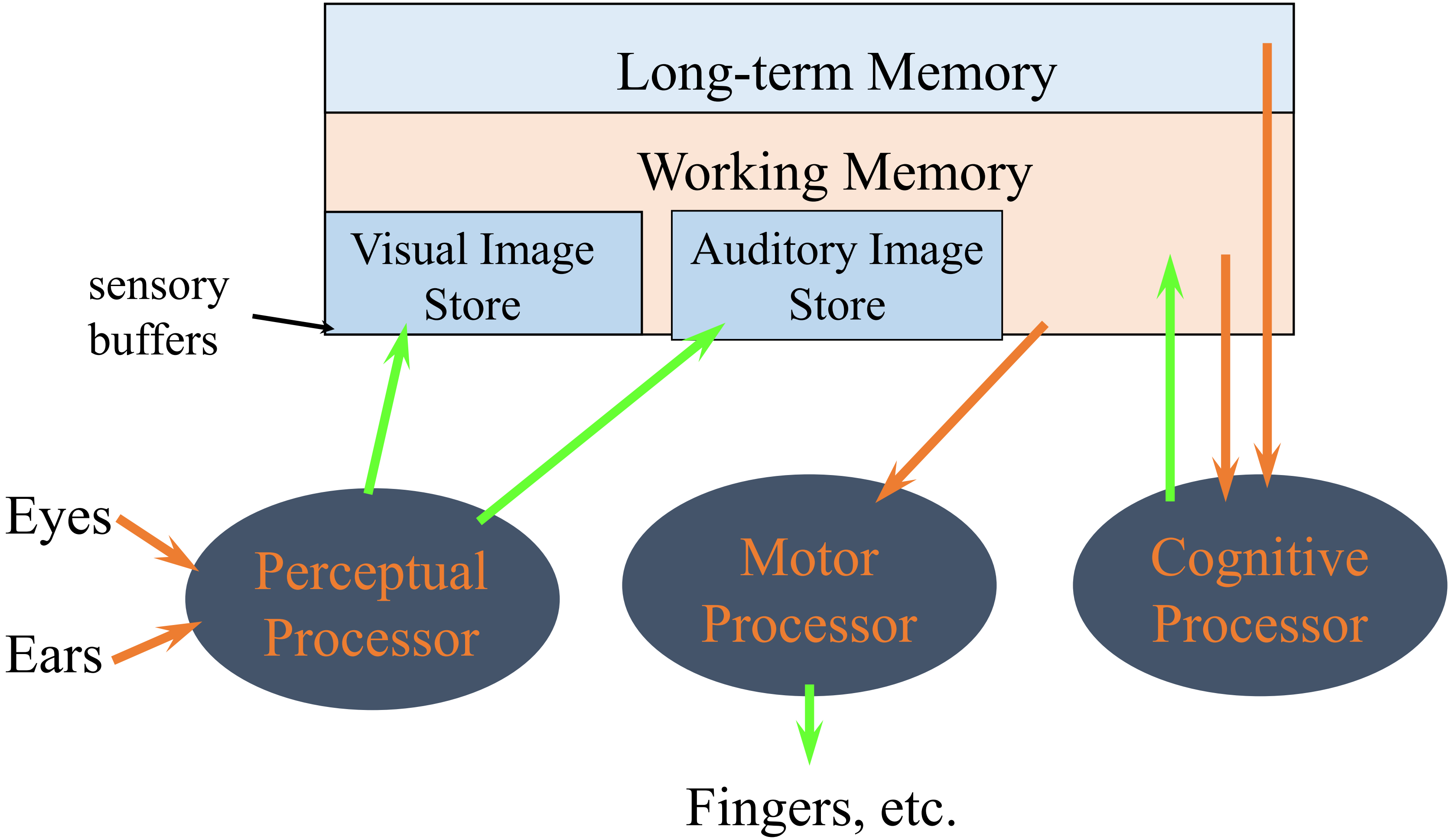
- “Crashtest Dummy for HCI"
- Family of techniques for modeling user’s interactions with a system
- Like cognitive walkthrough, begin with a set of tasks and the steps to accomplish them.

Cog Sci based Modeling

- Use what we know about human information processing in the brain to make predictions about how people will perform.
- Problem: the brain is still poorly understood.
  - Experts only allows us to get away from trying to predict how someone might misinterpret a UI element, etc.

The Model Human Processor

- Card, Moran, & Newell ('83) (based on empirical data)



Where are we?

- We can’t accurately predict planning time.
- We can predict execution time given an expert who makes no errors.

Which leaves....

- Learning
- Errors
- Long term recall
- Fatigue
- Etc.

Novice vs. Expert Use

- User testing focuses almost exclusively on first experiences.
  - Learning issues
  - Comprehension issues
- The GOMS family allows you to look at the expert experience.
  - Where will users time go? Is that where we expect?
  - Are there repetitive aspects we can eliminate?

So today…

- GOMS techniques are most useful for systems where
  - There will be experts
  - Users repeatedly perform a (relatively) small number of tasks
- GOMS is good for streamlining the efficiency of a process

How to do a GOMS Analysis

- Generate task description
  - Pick high-level user Goal
  - Write Method for accomplishing Goal - may invoke subgoals
  - Write Methods for subgoals
    - This is recursive
    - Stops when Operators are reached
- Evaluate description of task
- Apply results to UI
- Iterate

Adapted from Chris Long

Keystroke-Level Model

- Model was developed to predict time to accomplish a task on a computer
- Predicts expert error-free task-completion time with the following inputs:
  - a task or series of subtasks
  - method used
  - command language of the system
  - motor-skill parameters of the user
  - response-time parameters of the system
- Prediction is the sum of the subtask times and overhead

KLM Accuracy

- Widely validated in academia
- KLM predictions are generally within 10-20% of actual expert performance
- Simplified cognitive model

KLM

Keystroke level model

1. Predict

Action 1		x sec.
Action 2		y sec.
Action 3	+	z sec.
		t sec.

2. Evaluate

Time using interface 1

Time using interface 2

Symbols and values

Operator	Remarks	Time (s)
K	Press Key	0.2
B	Mouse Button Press	.10/.20
P	Point with Mouse	1.1
H	Home hand to and from keyboard	0.4
D	Drawing - domain dependent	-
M	Mentally prepare	1.35
R	Response from system - measure	-

Raskin excludes

Assumption: expert user

Raskin's rules

K	0.2
B	.10/.20
P	1.1
H	0.4
D	-
M	1.35
R	-

Rule 0: Initial insertion of candidate M's

M before K
M before P iff P selects command

i.e. not when P points to arguments

Rule 1: Deletion of anticipated M's

If an operator following an M is *fully* anticipated, delete that M.

e.g. when you point and click

Raskin's rules

Rule 2: Deletion of M's within cognitive units

If a string of MK's belongs to a cognitive unit, delete all M's but the first.

e.g. 4564.23

Rule 3: Deletion of M's before consecutive terminators

If a K is a redundant delimiter, delete the M before it.

e.g.)'

K	0.2
B	.10/.20
P	1.1
H	0.4
D	-
M	1.35
R	-

Raskin's rules

Chunking

Rule 2: Deletion of M's within cognitive units

If a string of MK's belongs to a cognitive unit, delete all M's but the first.

e.g. 4564.23

Rule 3: Deletion of M's before consecutive terminators

If a K is a redundant delimiter, delete the M before it.

e.g.)'

K	0.2
B	.10/.20
P	1.1
H	0.4
D	-
M	1.35
R	-

Raskin's rules

Rule 4: Deletion of M's that are terminators of commands

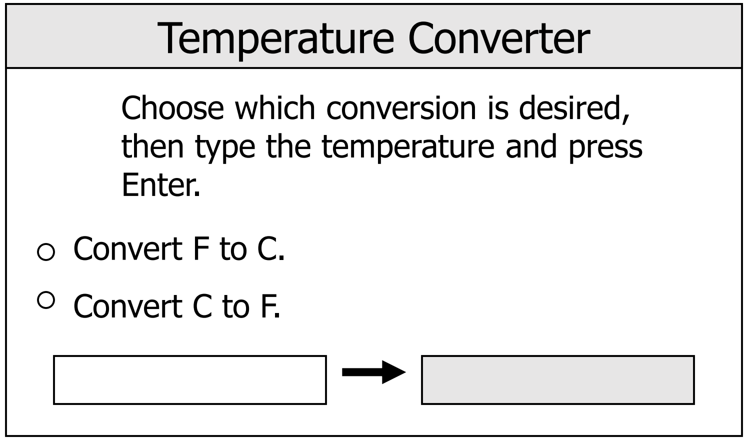
If K is a delimiter that follows a constant string, delete the M in front of it.

Rule 5: Deletion of overlapped M's

Do not count any M that overlaps an R.

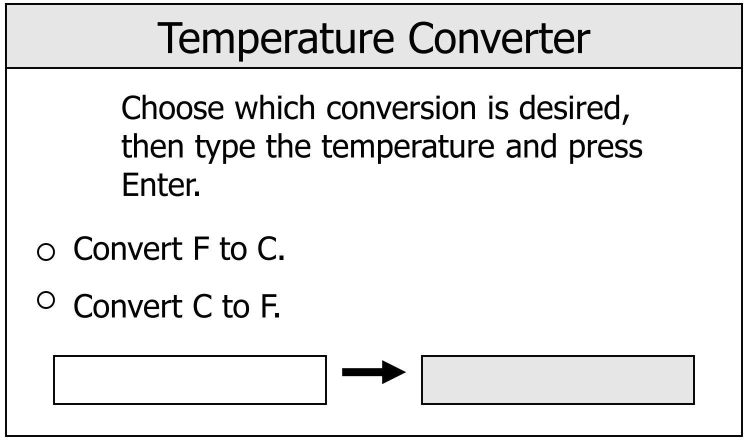
K	0.2
B	.10/.20
P	1.1
H	0.4
D	-
M	1.35
R	-

Example 1



K	0.2
B	.10/.20
P	1.1
H	0.4
D	-
M	1.35
R	-

Example 1



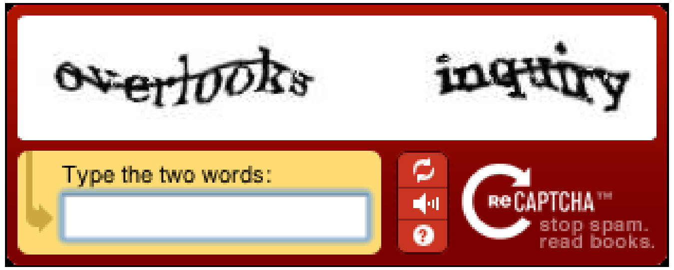
K	0.2
B	.10/.20
P	1.1
H	0.4
D	-
M	1.35
R	-

HPB (select F to C) PB (click in text box) HKKKKK	Apply Rule 0
HMPMB PMB HMKMKMKMK MK	Apply Rules 1 and 2
HMPB PB HMKKKKMK	Convert to numbers

.4+1.35+1.1+.20+ 1.1 + .2 +.4+1.35+4(.2)+1.35+.2

=8.45

Which is more efficient?



Operator	Description	Time (2)
K	Key Press	0.2
B	Mouse Button Press	0.1
P	Pointing with Mouse	1.1
H	Home hand to/from keyboard	0.4
M	Mentally prepare	1.35

How to do a GOMS Analysis

- Generate task description
- Evaluate description of task
- Apply results to UI
  - Look for ways to remove steps (learning + execution)
  - Look for ways to reuse sub-methods (learning)
  - Make sure that the end state is the goal (error prevention)
- Iterate

Adapted from Chris Long

Real World Example

- Nynex estimates that for every second saved in operator support, a company could save $3 million a year.
- Phone co. considering replacing old workstations with new ones.
  - 1000 new stations, $10K each.
  - Promised to reduce operator support time.

Toll and Assistance Operators

- The people you get when you dial 0
  - (person to person, collect calls, calling card calls)
- Their task – answer 3 questions:
  - Who pays?
  - At what rate?
  - Is the connection complete?

Field Trial to finalize purchase

- $70 million purchase for new system
- Wanted to gain in-house experience with training, using and maintaining new system
- Hoped to be able to show benefits of new system.

Prediction: New System Faster

- Based on –
  - Reduced keystrokes on most call categories
  - New system 880 msec faster at displaying a screenful of info
  - Estimated savings of $12.3 million a year

Alongside: GOMS Modeling

- Models being built during the field trial.
- Based on the system specs, not field observation.

Field Trial Data

- New system ~1 second slower
- Learning does not appear to be a factor
  - Differences in average times didn’t converge over time, as you would expect if there was a training effect.

Field Trial Results

- 78,240 phone calls, four months
- 24 TAOs using new system
- 24 TAOs using old system

What’s the problem?

- Original model captured a serial sequence of operations.
- In fact, TAOs do several things in parallel
  - Listening to a customer
  - Viewing display
  - Entering info via the keyboard
- Model needs to use CPM GOMS (Critical Path Method/Cognitive Perceptual Motor)

With CPM-GOMS

- Predicted .6 sec slower
  - Empirical data .8 sec slower

- Correctly predicted direction of difference in 13/15 call categories

GOMS – old vs. new

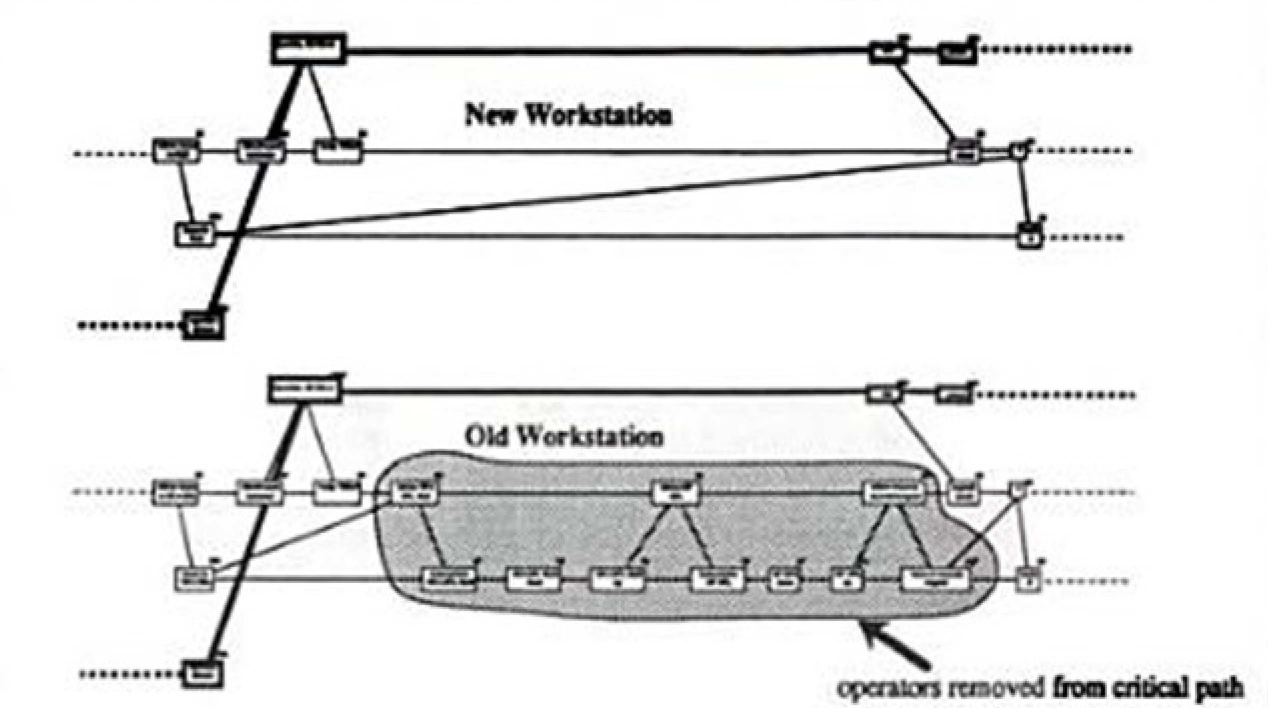


Figure 1. Section of GOMS analysis from near the beginning of the call. Notice that the new workstation (top) has removed two keystrokes (which had required 7 motor and 3 cognitive operators) from this part of the call. However, none of the ten operators removed were along the critical path (shown in bold). (Note: the print in the boxes is intentionally illegible.)

[Gray et al., GOMS Meets the Phone Company, Interact, 1990]

GOMS – old vs. new

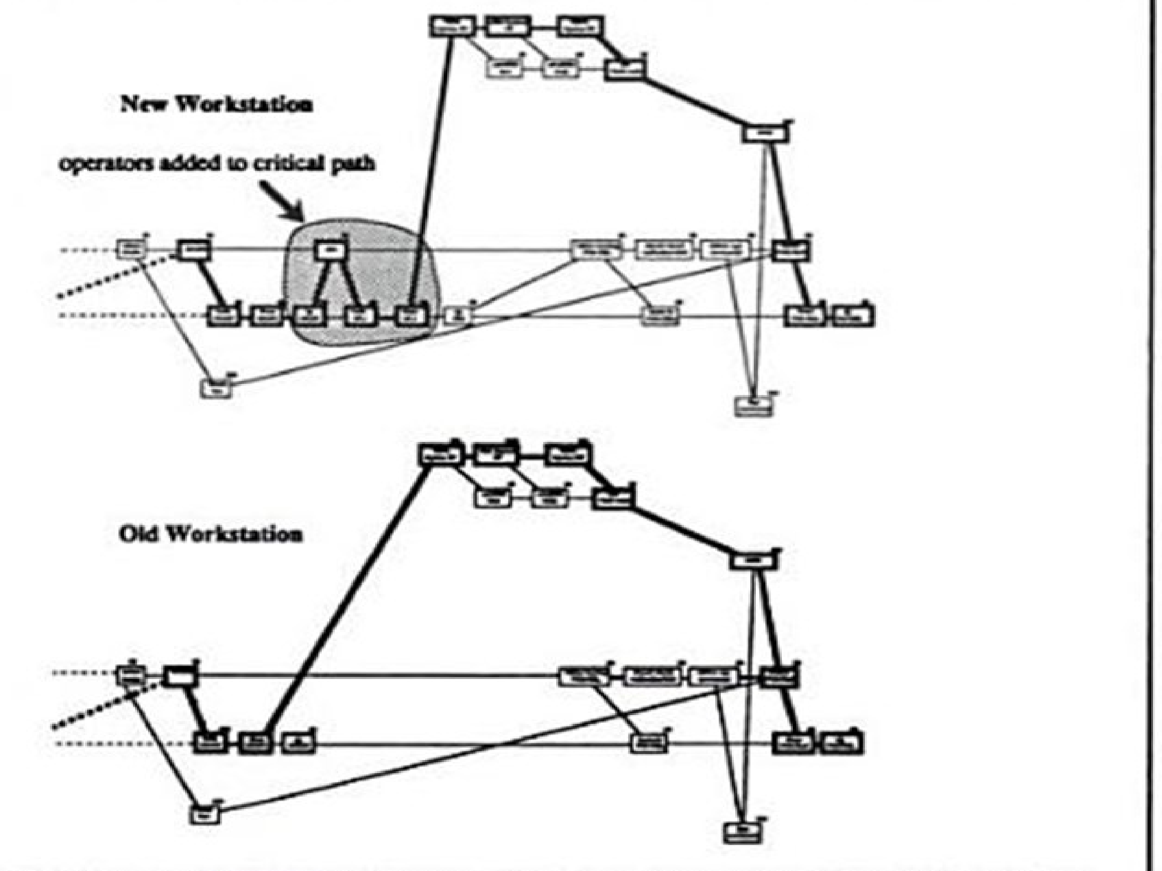


Figure 2. Section of GOMS analysis from the end of the call. Notice that the new workstation (top) has added one keystroke to this part of the call which results in four operators (three motor and one cognitive) being added to the critical path (shown in bold). (Note: the print in the boxes is intentionally illegible.)

[Gray et al., GOMS Meets the Phone Company, Interact, 1990]

Model as Explanation

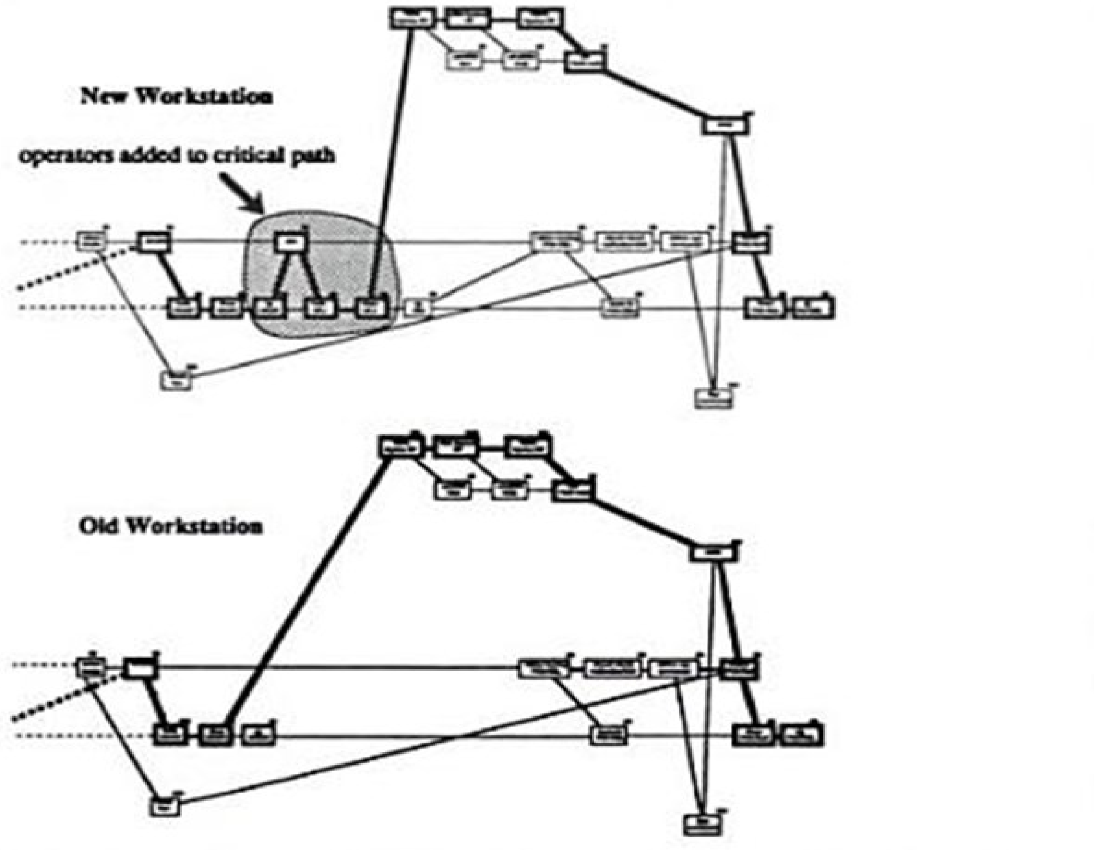


Figure 2. Section of GOMS analysis from the end of the call. Notice that the new workstation (top) has added one keystroke to this part of the call which results in four operators (three motor and one cognitive) being added to the critical path (shown in bold). (Note: the print in the boxes is intentionally illegible.)

[Gray et al., GOMS Meets the Phone Company, Interact, 1990]

Where do I use this?

- Situations where task performance is critical
  - Airline/auto displays
  - Emergency management systems
  - Process control systems
  - Customer service systems
  - Etc.

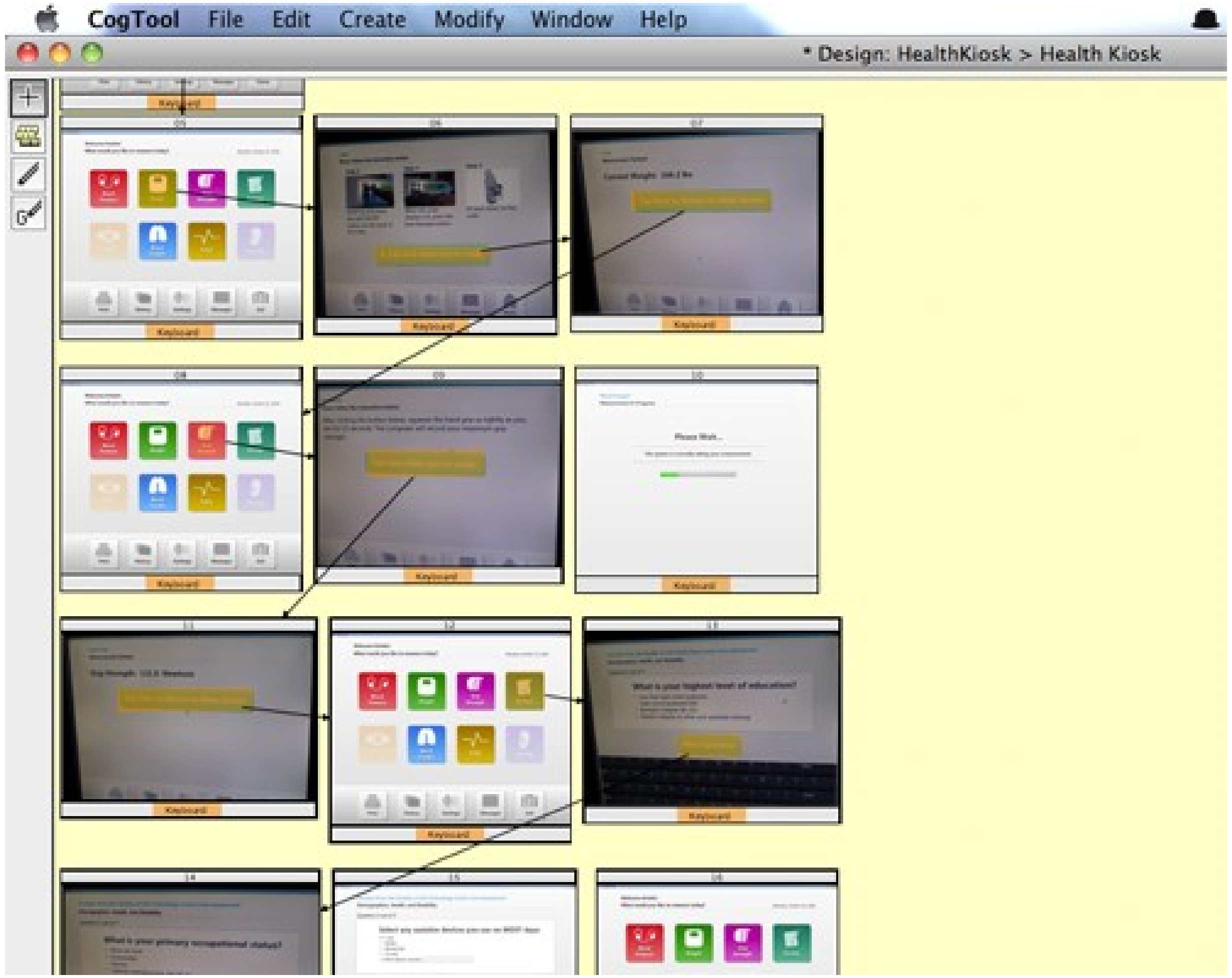
Caveats

- *Very* idealized
  - Assuming expert user
  - Assuming perfect performance
  - Depending on the scenario, you may need to choose your version of GOMS carefully

CogTool - Automating KLM

- Free software from Bonnie John's group at CMU
- Take a collection of images
- Define the transitions
- Out comes a KLM model

Health Kiosk Screen Shots



From: http://www.andrew.cmu.edu/user/sanchitg/healthkiosk.html

Example CogTool Comparison Results

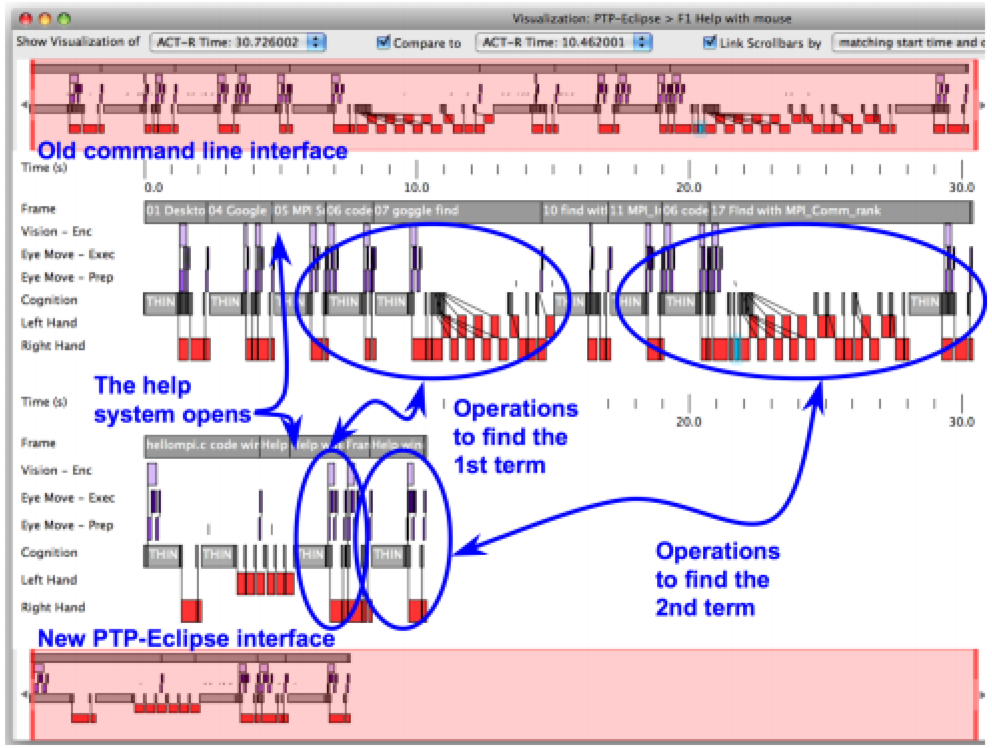


Figure 3. Timeline visualization showing the key difference between the two systems.

From Bellamy, John, and Kogan Deploying CogTool: Integrating Quantitative Usability into Real-World Software Development

Constant Time for Pointing

Operator	Remarks	Time (s)
K	Press Key	0.2
B	Mouse Button Press	.10/.20
P	Point with Mouse	1.1
H	Home hand to and from keyboard	0.4
D	Drawing - domain dependent	-
M	Mentally prepare	1.35
R	Response from system - measure	-

Can we do better? → P

Assumption: expert user

Fitts' Law

Models <u>movement time</u> for <u>selection tasks</u>

The <u>movement time</u> for a <u>well-rehearsed</u> selection task

- increases as the <u>distance</u> to the target increases
- decreases as the <u>size</u> of the target increases

Fitts' Law

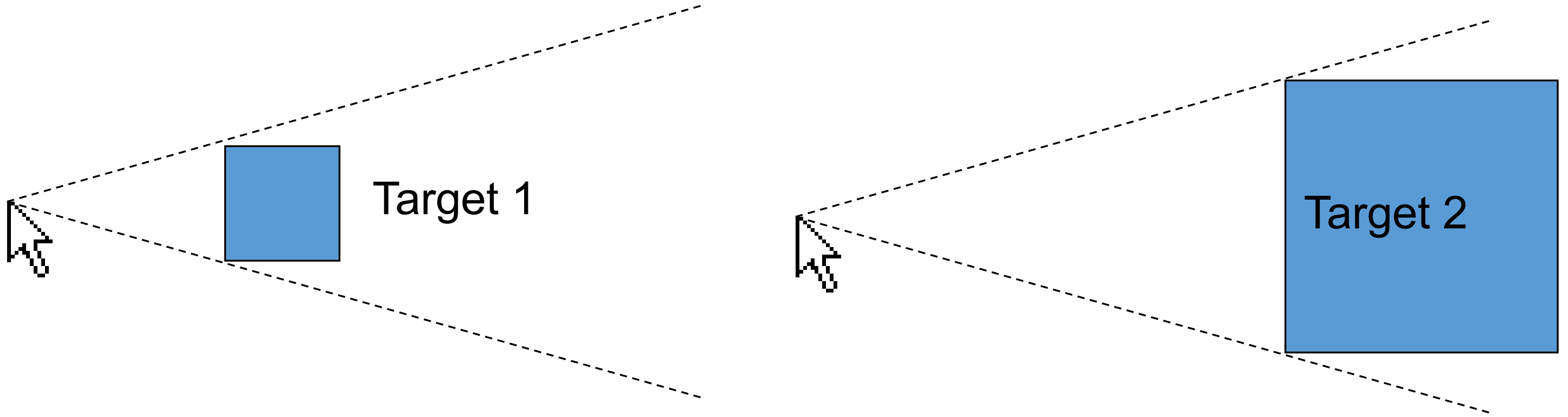
Time (in msec) = a + b $\log_2(D/S+1)$

where
a, b = constants (empirically derived)
D = distance
S = size

ID is Index of Difficulty = $\log_2(D/S+1)$

Fitts' Law

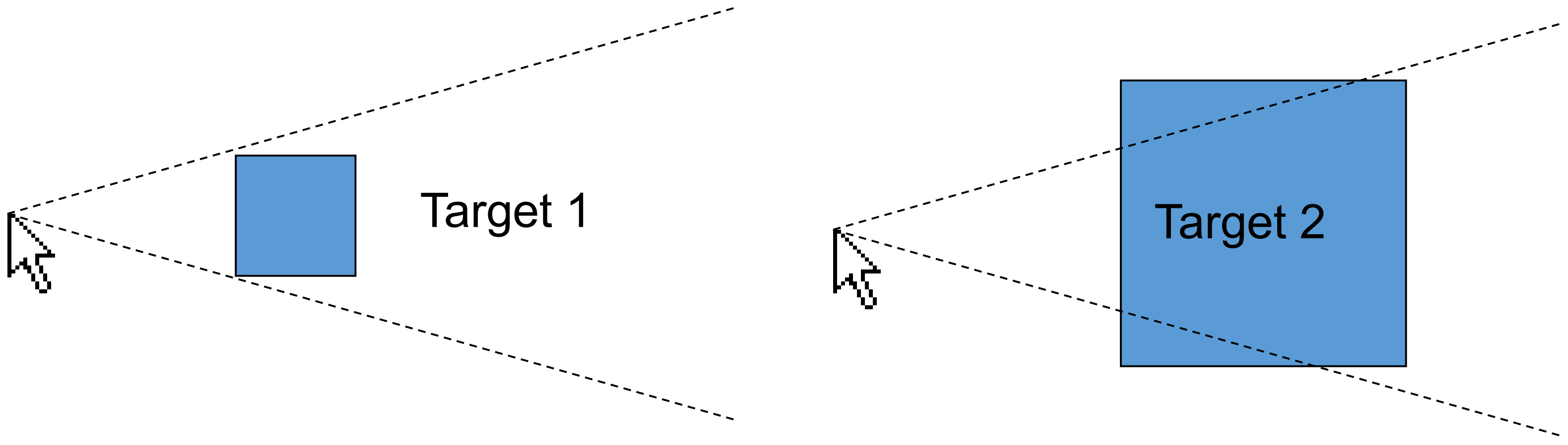
$$\text{Time} = a + b \log_2(D/S+1)$$



Same ID → Same Difficulty

Fitts' Law

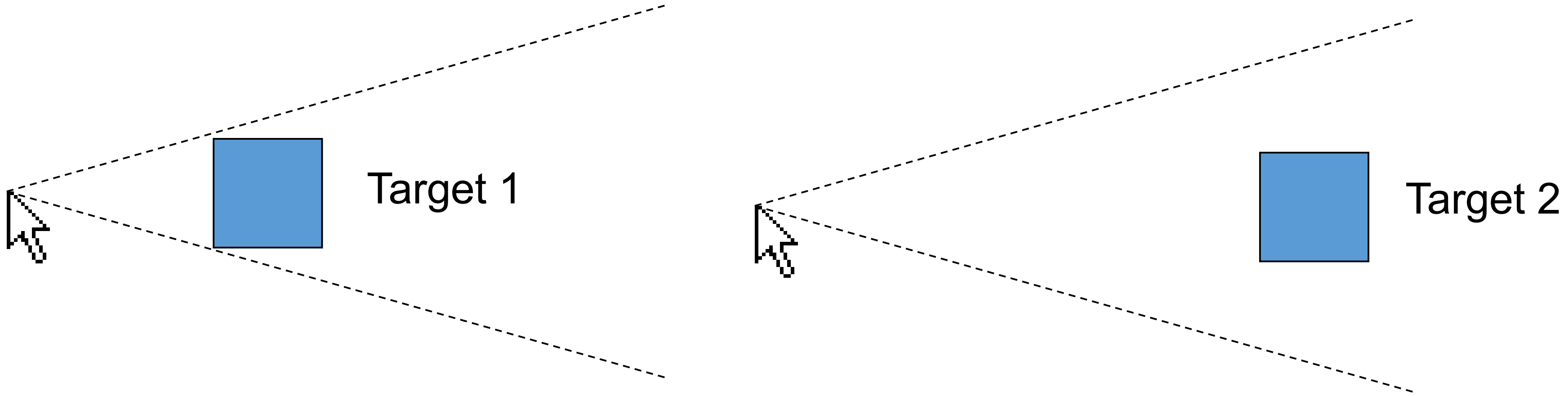
$$\text{Time} = a + b \log_2(D/S+1)$$



Smaller ID → Easier

Fitts’ Law

$$\text{Time} = a + b \log_2(D/S+1)$$



Larger ID → Harder

Determining Constants for Fitts' Law

- To determine a and b design a set of tasks with varying values for D and S (conditions)

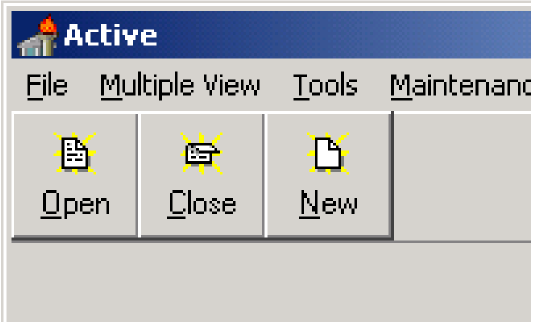
- For each task condition
  - multiple trials conducted and the time to execute each is recorded and stored electronically for statistical analysis

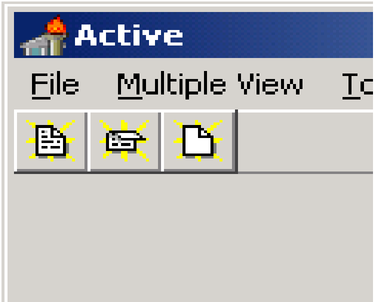
- Accuracy is also recorded
  - either through the x-y coordinates of selection or
  - through the error rate — the percentage of trials selected with the cursor outside the target

A Quiz Designed to Give You Fitts

- *http://www.asktog.com/columns/022DesignedToGiveFitts.html*

- *Microsoft Toolbars offer the user the option of displaying a label below each tool. Name at least one reason why labeled tools can be accessed faster. (Assume, for this, that the user knows the tool and does not need the label just simply to identify the tool.)*





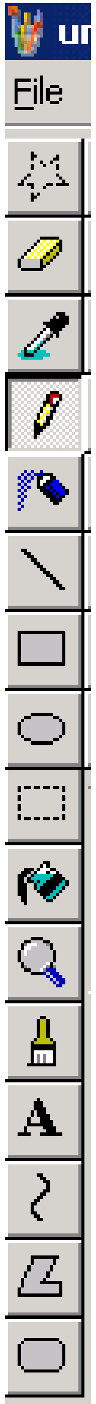
A Quiz Designed to Give You Fitts

1. The label becomes part of the target. The target is therefore bigger. Bigger targets, all else being equal, can always be acccessed faster. Fitt's Law.

2. When labels are not used, the tool icons crowd together.

A Quiz Designed to Give You Fitts

- *You have a palette of tools in a graphics application that consists of a matrix of 16x16-pixel icons laid out as a 2x8 array that lies along the left-hand edge of the screen. Without moving the array from the left-hand side of the screen or changing the size of the icons, what steps can you take to decrease the time necessary to access the average tool?*



A Quiz Designed to Give You Fitts

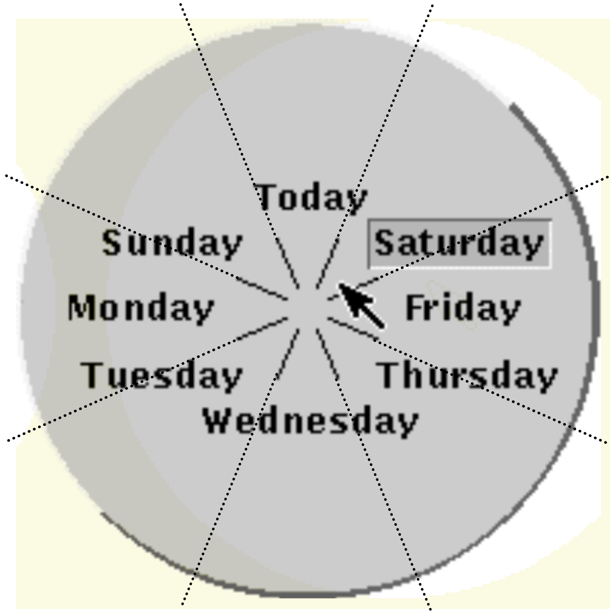
1. Change the array to 1X16, so all the tools lie along the edge of the screen.

2. Ensure that the user can click on the very first row of pixels along the edge of the screen to select a tool. There should be no buffer zone.

Fitts' Law Example

Pop-up Linear Menu

Pop-up Pie Menu



- Which will be faster on average?
  - pie menu (bigger targets & less distance)

Principles of Operation (cont.)

- Fitts' Law
  - moving hand is a series of microcorrections
    - correction takes $T_p + T_c + T_m$ = 240 msec
  - time T_{pos} to move the hand to target size S which is distance D away is given by:
    - $T_{pos} = a + b \log_2 (D/S + 1)$
  - summary
    - time to move the hand depends only on the *relative precision* required